APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/008,864	12/06/2001	Cary Lee Bates	CA920010004US1	5046

7590   10/03/2007

Grant A. Johnson
IBM Corporation
Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

EXAMINER
ROMANO, JOHN J

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
10/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Office Action Summary	10/008,864	BATES ET AL.
	Examiner	Art Unit	
	John J. Romano	2192	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>08 March 2007 and 18 June 2007</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1,2,4,5,7-12 and 25-32</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1, 2, 4, 5, 7-12 and 25-32</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

Remarks

1.   In view of the appeal brief filed on June 18th, 2007, PROSECUTION IS HEREBY

REOPENED. A new grounds of rejection is set forth below.

To avoid abandonment of the application, appellant must exercise one of the following

two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37

CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an

appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee

can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have

been increased since they were previously paid, then appellant must pay the difference between

the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing

below:

Additional Comments:

The examiner notes that the triggering expression event, wherein an expression "represents a non-executable data value" itself required to have "a state".

Correspondingly, as presently claimed in claim 1 and other independent claims, the "trigger expression" is now interpreted to require **both** a data value and a state/status of the data value (emphasis added). This is different from Examiner's prior position that a variable by itself "represents a non-executable data value [having a state]".

The trigger expression, which is interpreted as explicitly defined in Applicant's specification, to cover an activity to an addressable memory location used by a computer program, intended to include L values, or values that can expressed as being left of an equal sign in a logical or arithmetic expression (such as a high level language construct), as argued by Applicant and reflected in the specification, with explicit examples (page 4, paragraph [0038]):

"Examples of expressions may include an array element, a pointer expression, a substring expression, etc., as well as variables."

In this interpretation the L-value is interpreted as the non-executable data value and the instance of that value is interpreted as the state.

Alternatively, it is note that the plain language of the claim, in particular "state" could be interpreted to mean a mode or status of the trigger expression. From this perspective the state may be interpreted to mean Read and/or Write.

For the sake of compact prosecution, the examiner is interpreting the state (instance value, i.e., instance value of specified element, such as particular array element/variable/ pointer

expression at time of reference or L-value) as an instance of a non-executable data value

(memory location represented by the L-value above). Here, the L-value represents the specified

non-executable data value, wherein a respective detected occurrence of an access

(read/write/reference) to said location in memory containing a particular instance value (state)

represents the trigger expression to create a snapshot (history of particular values at time of

reference) corresponding to the occurrence of an access to the location of memory.

Accordingly, the prosecution is reopened and now reflects the "trigger expression"

respectively as provided herein-below in the claim rejections.

Claim Rejections

Claims **1, 2, 4, 5, 7-12** and **25-32**, are pending claims, stand rejected in light of the claim

rejections below.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1.   Claims **1, 2, 5** and **7** are rejected under 35 U.S.C. 103(a) as being unpatentable over

Lindsey, US 5,896,536 (hereinafter **Lindsey**).

In regard to claim **1, Lyndsey** discloses:

- *"A method of tracing the activity of an expression, said method comprising the machine-implemented steps of ..."* (E.g., see Figure 2 & Column 2, lines 45-49), wherein a technique for debugging object oriented program code with tracepoints which trigger tracing based upon access of or modification to a specific data component, such as instance variables in Smalltalk language programs (i.e., a high level language construct) as exemplified expressly (See Col. 1, lines 55-56).

- *"...(b) receiving, from a user, a specification of the trigger expression to be traced in the machine-implemented process..."* (E.g., see Figure 6 (202 + 206) & Column 8, lines 11-14), wherein step 202 determines if the method (designated data component) was specified as a tracepoint access method (or in other words specified to be traced). Next step 206, evaluate the trigger parameters (conditions) to determine if the type of access and conditions trigger or affect tracing (E.g., resulting in L value or read/write/reference, see column 8, lines 28-31).

- *"...said trigger expression representing a non-executable data value having a state..."* (E.g., see Figure 6 (210) & Column 8, lines 49-57), wherein different types of access (e.g., read/write) are set by the developer via window 80, corresponding to representations of non –executable data values (i.e., locations of memory represented by an array element, a pointer expression, a substring expression, variables, etc.), wherein the state is the instance of

(snapshot) the particular location of memory at the time of reference

represented by the trigger expression.

- *"...(c) responsive to steps (a) and (b), monitoring execution of said machine-*

  implemented process to detect occurrences of a plurality of references to a

  location in machine memory representing a state of said trigger expression,

  wherein each said occurrence of a reference to a location in machine memory

  representing a state of said trigger expression occurs as a result of executing

  said machine-implemented process..." (E.g., see Figure 5 & Column 6, lines

  35-40), wherein the developer can set tracing for specific data components or

  instance variables within the object oriented program being debugged

  (Column 6, lines 47-59), wherein tracing is set relative to a type (read/write)

  of access, and specific conditions (i.e., the first, tenth, eightieth) during

  execution of the object will cause tracing to be turned on and off.

- *"...(d) responsive to each detected occurrence of a reference to said location*

  in machine memory representing a state of said trigger expression, storing the

  respective state of the trigger expression at the time of the respective detected

  occurrence of a reference to said location in machine memory representing a

  state of said trigger..." (E.g., see Figure 2 & Column 5, line 62 – Column 6,

  line 8), wherein read and/or write access methods 56, 58 are added to the

  object 46 such that any attempt to access the designated instance variable 54

  of the object 46 (detected occurrence) must be routed through one of the

  access methods 56, 58. Depending upon the parameters set by the developer a

trace message is sent to a trace manager 60 of a debugger 62 for implementing

tracing and the desired tracing relative to the designated instance variable 54

of the object 46 is carried out.

- "...*to create a history of said trigger expression within the machine-*

  implemented process, said storing step being performed without interrupting

  the machine-implemented process..." (E.g., see Figure 2 & Column 6, lines 8-

  12), wherein the methods 56, 58 provide read and write access to the

  designated instance variable 54 during execution. Then prior to ending

  execution the methods 56, 58 may further effect tracing after performing the

  appropriate access of the designated instance variable 54 (E.g., sending a

  second trace method) and as a result store the values of the specified locations

  (snapshot).

- "...*(e) restoring the state of the trigger expression when requested.*" (E.g., see

  Figure 5 (90) & Column 6, lines 65-67), wherein the developer may request

  the trace option to be restored after access.

But **Lyndsey** does not expressly disclose "...*(a) receiving, from a user, a specification of*

a machine-implemented process in which a trigger expression is to be traced..." However,

Lyndsey discloses (E.g., see Figure 5 & Column 2, lines 45-49), wherein a user specifies a data

component of an object oriented program (machine- implemented process) undergoing

debugging. Furthermore, **Lyndsey** discloses (Figure 2, & Column 5, lines 49-54) the program

40 comprising a plurality of objects (E.g., see objects 46 + 48). Therefore, it would have been

obvious, to one of ordinary skill in the art at the time the invention was made, to specify a

particular program 40 to be debugged as taught by **Lyndsey** in order to trace the desired relevant

information (See Column 1, line 62 – Column 2, line 2).

In regard to claim **2**, the rejections of base claim **1** are incorporated. Furthermore,

Lindsey discloses:

- "...*(a) imposing a condition onto the trigger expression; and (b) storing the*

    state of the trigger expression only when the condition is satisfied." (E.g., see

    Figure 5 & Column 6, lines 59-61), wherein specific conditions can be

    identified by the developer under which the trace function is to be initiated (or

    storing) by setting an appropriate condition in box 86.

In regard to claim **5**, the rejections of base claim **1** are incorporated. Furthermore,

Lyndsey discloses:

- "...*results in an L value during the machine-implemented process.*" (E.g., see

    Column 2, lines 44-47), wherein triggering on selective events, such as

    accessing memory locations (read or write) is disclosed.

In regard to claim **7**, the rejections of base claim **1** are incorporated. Furthermore,

Lyndsey discloses::

- "...*reference to said location in machine memory representing a state of said*

    trigger expression is a Read and/or a Write." (E.g., see Column 2, lines 44-

    47), wherein triggering on selective events, such as accessing memory

    locations (read or write) is disclosed.

2.   Claims **4, 8-12** and **25-32** are rejected under 35 U.S.C. 103(a) as being unpatentable over

Lyndsey in view of **Wygodny**, US 6,282,701 (hereinafter **Wygodny**).

In regard to claim **4**, the rejections of base claim **1** are incorporated. But **Lyndsey** does

not expressly disclose "...*(a) displaying the history such that the state of the trigger expression*

each time the trigger expression was active can be displayed separately." However, **Wygodny**

discloses:

> - "...*(a) displaying the history such that the state of the trigger expression each*
>
>   *time the trigger expression was active can be displayed separately.*" (E.g., see
>
>   Figure 1C & Column 8, lines 15-20), wherein the data stored or history is
>
>   displayed according to filters set by the user allowing the user to display a
>
>   particular triggered expression separately if desired.

Lyndsey and **Wygodny** are analogous art because they are both concerned with the same

field of endeavor, namely, tracing the execution path of a computer object of an object oriented

program. Therefore, at the time the invention was made, it would have been obvious to a person

of ordinary skill in the art to combine **Wygodny's** teaching of "...*(a) displaying the history such*

that the state of the trigger expression each time the trigger expression was active can be

displayed separately." with **Lyndsey's** tracing method. The motivation to do so would have

been to monitor the execution of the code based on selections or options from the user as

suggested by **Wygodny** (Column 3, lines 7 –12), wherein the conditional trigger or expression

would provide the user with further options. Furthermore, **Lyndsey** suggests "...the ability to

trigger the generation of trace data based upon a specific data component so that information can

be obtained relative to the data component during execution of logic units would be a valuable

tool for developers in the debugging of programs". Thus it would have been obvious, to a person

of ordinary skill in the art to separately display the state of the active trigger selection with

Lyndsey's tracing method.

In regard to claim **8**, the rejections of base claim **1** are incorporated. Furthermore,

Lyndsey discloses:

- *"...(a) receiving, from a user, a specification of at least one attached*

   expression; (b)...storing the respective state of the at least one attached

   expression ... within the machine-implemented process..." (E.g., see Figure 4

   & Column 7 lines 3-10), wherein the developer designates one or more

   instance variables to which the triggering of tracing is desired (Step 100), the

   developer designates specific functional parameters (e.g., read/write) for the

   trace function via the window 70 of Fig. 5. The specific functional parameters

   may be the same for all of the designated instance variables of a particular

   object (thereby being attached by the trigger expression), or may be set

   specifically for each designated instance variables.

But **Lyndsey** does not expressly disclose *"...(a) displaying the history such that the state*

of the trigger expression each time the trigger expression was active can be displayed

separately." However, **Wygodny** discloses:

- *"...(c) restoring the state of the at least one attached expression when*

   requested." (E.g., see Figure 3A & Column 18, lines 30-43), wherein the

   developer can choose any arguments, or return values, thereby storing the

   state of a chosen function and respective attached expressions or variables,

wherein trace can then be displayed according to the developers choice

(restoring) when requested.

- "... *at the time of the respective detected occurrence of a reference to said*

   location in machine memory representing a state of said trigger expression,

   the states of the at least one attached expression being associated with said

   history of said trigger expression..." (E.g., see Figure 1C & Column 26, lines

   33-34), wherein the variable or expression to be recorded (history - snapshot)

   and corresponding attached variables are useful when used dynamically to

   allow what is to be traced or was traced relative to an access (read/write)

   associated with said history of trigger expression.

In regard to claim **9**, the rejections of base claim **1** are incorporated. Furthermore,

Lindsey discloses:

- "*...the machine-implemented process is a computer program.*" (E.g., see

   Figure 2 (140)).

In regard to claims **10** and **11**, the rejections of base claim **1** are incorporated.

Furthermore, **Lindsey** discloses:

- "*...included in an object level trace program.*" (E.g., see Figure 2 & Column

   2, lines 45-47), wherein an object oriented debugging program is disclosed.

In regard to claim **12**, **Lyndsey** discloses:

- "*A method of tracing the activity of an expression in an executing computer*

   program..." (E.g., see Figure 2 & Column 2, lines 45-49), wherein a

   technique for debugging object oriented program code with tracepoints which

trigger tracing based upon access of or modification to a specific data

component, such as instance variables in Smalltalk language programs (i.e., a

high level language construct) as exemplified expressly (See Col. 1, lines 55-

56).

- "...(a) *receiving, from a user, a specification of the computer program in*

   which a trigger expression resulting in an L value during the execution of the

   computer program is to be traced..." (E.g., see Figure 4 & Column 2, lines

   44-47), wherein triggering on selective events, such as accessing memory

   locations (read or write) is disclosed.

- "...(b) *a specification of the trigger expression and any optional attachment*

   expressions to be traced in the computer program...." (E.g., see Figure 4 &

   Column 7, lines 3-10), wherein the specific functional parameters may be the

   same for all of the designated instance variables, or may be set specifically for

   each designated instance variables. Figure 4, (72) and related text teach,

   optional attachment expressions (e.g, instance variables/class variables,etc.)

   may be selected. Also, note (Column 5, lines 30-34), wherein tracing may be

   triggered by associating tracepoints with instance variables, class variables or

   class instance variables (referred to collectively as instance variables

   hereinafter).

- "...*(c) imposing a condition onto the trigger expression...* " (E.g., see Figure

   6 & Column 8, lines 39-48), wherein a tracing operation is stored when a

   predetermined condition is detected (satisfied).

- *"...(e) responsive to each detected occurrence of an access to said location in memory containing a value representing said trigger expression, if said condition is satisfied, then storing the respective state of the trigger expression..."* (E.g., see Figure 5 & Column 6, lines 47-61), wherein relative to the selected instance variables access (i.e., read/write occurrence to memory of trigger expression), specific conditions can be identified by the developer under which the trace function is to be initated by setting an appropriate condition in box 86 causing tracing the state/instance value to be stored responsive to a memory containing a value being accessed.

- *"...and any optional attachment expressions at the time of the respective detected occurrence of an access to said location in memory containing a state representing the trigger expression to create a snapshot corresponding to the respective detected occurrence of an access to said location in memory..."* (E.g., see Figure 5 & Column 5, lines 35-38) wherein the triggering of the trace function causes the debugger to collect trace information regarding the present status of items related to the program and what is occurring during the execution. It is noted that the same functional trigger set to one or more variables/data components of an object reference (e.g, see Figure 3 & 7, lines 3-10), result in the attached trigger expressions.

But **Lyndsey** does not expressly disclose *"...(g) displaying the profile such that each snapshot can be displayed separately; and (h) restoring the state of each snapshot, when requested."* However, **Wygodny** discloses:

- *"...(f) creating a profile of the trigger expression comprising storing each snapshot; (g) displaying the profile such that each snapshot can be displayed separately; and (h) restoring the state of each snapshot, when requested."* (E.g., see Figure 3A & Column 18, lines 30-43), wherein the developer can choose any arguments, return values and selected source lines, thereby storing the state of a chosen function and attached expressions or variables (profile), wherein trace can then be displayed according to the developers choice (restoring) when requested.

Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to impose a trigger condition on **Wygodny's** tracing. The motivation to do so would have been to monitor the execution of the code based on selections or options from the user as suggested by **Wygodny** (Column 3, lines 7 –12), wherein the conditional trigger or expression would provide the user with further options. Furthermore, **Lyndsey** suggests "...the ability to trigger the generation of trace data based upon a specific data component so that information can be obtained relative to the data component during execution of logic units would be a valuable tool for developers in the debugging of programs". Thus it would have been obvious, to a person of ordinary skill in the art to include **Wygodny's** tracing technique with **Lyndsey**'s tracing method. See Claim 1 for remaining limitations.

In regard to claim **25, Lyndsey** discloses:

- *"An article of manufacture, comprising a data storage medium tangibly embodying a program of machine readable instructions executable by an*

electronic processing apparatus..." (E.g., see Figure 1 & Column 4, lines 48-
61).

- "...*(a) initiating a user interface to exchange data input/output with a user
  and an electronic processing apparatus...*" (E.g., see Figure 5 & Column 6,
  lines 47-50), a user interface for qualifying a tracepoint.

- "...*(d) causing the electronic processing apparatus to execute the identified
  program ...*" (E.g., see Figure 6, element 200), wherein the program is
  executed.

But **Lyndsey** does not expressly disclose "...*(g) maintaining the capability to restore
each snapshot and display each snapshot to the user.*". However, **Wygodny** discloses:

- "...*(g) maintaining the capability to restore each snapshot and display each
  snapshot to the user.*" (E.g., see Figure 1C & Column 8, lines 15-20), wherein
  the data stored or history is displayed (restored) according to filters set by the
  user allowing the user to display a particular triggered expression separately if
  desired.

But **Lyndsey** and **Wygodny** do not expressly disclose "...*(b) requesting ...a trigger
expression...*". However, it would have been obvious to one of ordinary skill in the art, at the
time the invention was made to request a trigger expression from a user. The motivation to do
so was provided by **Wygodny** (E.g., see Figure 5 & Column 13, lines 50-54), wherein the
developer may specify which variables or expressions to be traced (520), as addressed above.
Thus, it would have been obvious that the means to input the specification is the equivalent to
requesting the specification from the user.

Therefore, at the time the invention was made, it would have been obvious to a person of

ordinary skill in the art to embody **Lyndsey** and **Wygodny's** tracing instructions on an article of

manufacture. The motivation to do so would have been to send the tracer program to the client

as suggested by **Wygodny** (Column 3, lines 30-32), wherein the developer would not need to

visit the remote site. Thus it would have been obvious, to a person of ordinary skill in the art to

include a **Wygodny's** tracing program on an article of manufacture. See claim 1 for remaining

limitations.

In regard to claim **26**, the rejections of base claim **20** are incorporated. Furthermore,

Lyndsey discloses:

- *"...requesting the user to assign conditions to the trigger expression*

   whereupon when the conditions are satisfied, a snapshot of the trigger

   expression is stored." (E.g., see Figure 5, (86) & Column 6, lines 59-61),

   wherein a tracing operation is stored when a predetermined condition is

   detected (satisfied), wherein the predetermined condition was input from the

   user (requested from the user) via the condition (Figure 5, block 86).

In regard to claim **27**, the rejections of base claim **25** are incorporated. Furthermore,

Wygodny discloses:

- *"...the user to indicate attached expressions whose states are also stored in a*

   corresponding snapshot whenever a snapshot is stored for the trigger

   expression." (E.g., see Figure 3A & Column 18, lines 30-43), wherein the

   developer can choose any arguments, return values and selected source lines,

   thereby storing the state of a chosen function and attached expressions or

variables (profile), wherein trace can then be displayed according to the

developers choice (restoring) when requested.

In regard to claims **28-30**, this is a digital data processing device version of the article of

manufacture claims that have been addressed in the above claims **25-27**, wherein all claimed

limitations have also been addressed and/or cited as set forth above.

In regard to claim **31**, the rejections of base claim **28** are incorporated. Furthermore,

Wygodny discloses:

- "...*execute on the same computer.*" (E.g., see Figure 1B & Column 5, lines

    37-53), wherein the device that does the tracing (trace library, (102)) is on the

    clients computer.

In regard to claim **32**, the rejections of base claim **28** are incorporated. Furthermore,

Wygodny discloses:

- "...*the first computer program and the second computer program execute on

    separate units connected by a data communications link.*" (E.g., see Figure 2

    & Column 6, lines 55-65), wherein the device that does the tracing (trace

    library, (124)) is separated from the digital logic device and connected by a
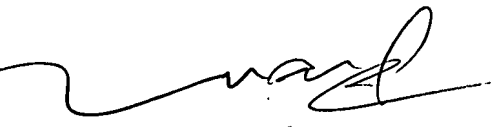
    data communications link as shown.

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to John J. Romano whose telephone number is (571) 272-3872. The

examiner can normally be reached on 8-5:30, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JJR

TUAN DAM
SUPERVISORY PATENT EXAMINER